Probabilistic Human-Computer Trust Handling

Florian Nothdurft[⋆], **Felix Richter**[†] **and Wolfgang Minker**[⋆]
[⋆]Institute of Communications Engineering
[†]Institute of Artificial Intelligence
Ulm University
Ulm, Germany
florian.nothdurft, felix.richter, wolfgang.minker@uni-ulm.de

Abstract

Human-computer trust has shown to be a critical factor in influencing the complexity and frequency of interaction in technical systems. Particularly incomprehensible situations in human-computer interaction may lead to a reduced users trust in the system and by that influence the style of interaction. Analogous to human-human interaction, explaining these situations can help to remedy negative effects. In this paper we present our approach of augmenting task-oriented dialogs with selected explanation dialogs to foster the human-computer trust relationship in those kinds of situations. We have conducted a web-based study testing the effects of different goals of explanations on the components of human-computer trust. Subsequently, we show how these results can be used in our probabilistic trust handling architecture to augment pre-defined task-oriented dialogs.

1 Introduction

Human-computer interaction (HCI) has evolved in the past decades from classic stationary interaction paradigms featuring only human and computer towards intelligent agent-based paradigms featuring multiple devices and sensors in intelligent environments. For example, ubiquitous computing no longer seems to be a vision of future HCI, but has become reality, at least in research labs and prototypical environments. Additionally, the tasks a technical system has to solve cooperatively with the user have become increasingly complex. However, this change from simple task solver to intelligent assistant requires the acceptance of and the trust in the technical system as dialogue partner and not only as ordinary service device.

Especially trust has shown to be a crucial part in the interaction between human and technical system. If the user does not trust the system and its actions, advices or instructions the way of interaction may change up to complete abortion of future interaction (Parasuraman and Riley, 1997). Especially those situations in which the user does not understand the system or does not expect the way how the system acts are critical to have a negative impact on the human-computer trust (HCT) relationship (Muir, 1992). Those situations do occur usually due to incongruent models of the system: During interaction the user builds a mental model of the system and its underlying processes determining system actions and output. However, if this perceived mental model and the actual system model do not match the HCT relationship may be influenced negatively (Muir, 1992). This may, for example, be due to a mismatch in the expected and the actual system action and output.

For example, if a technical system would assist the user in having his day scheduled in a time effective manner, the user would be in a vulnerable situation of relying on the reasoning capabilities of the system. However, when the user-expected time schedule does not match the system-generated, the question arises if the user will trust the system, despite lacking the knowledge if the schedule is correct. If the user trusts the automated day scheduling capability of the system, he will probably attend the appointments exactly as scheduled. However, if he does not trust this automated outcome he won't rely on it and will question the plan.

Therefore, the goal should be to detect those critical situations in HCI and to react appropriately. If we take a look at how humans detect and handle critical situations, we can conclude that they use contextual information combined with interpreted multimodal body analysis (e.g., facial expression, body posture, speech prosody) for detection and usually some sort of explanation to

Proceedings of the SIGDIAL 2014 Conference, pages 51–59,
Philadelphia, U.S.A., 18-20 June 2014. ©2014 Association for Computational Linguistics

Goals	Details
Transparency	How was the systems answer reached?
Justification	Explain the motives of the answer?
Relevance	Why is the answer a relevant answer?
Conceptualization	Clarify the meaning of concepts
Learning	Learn something about the domain

Table 1: Goals of explanation after (Sørmo and Cassens, 2004). These goals subsume different kinds of explanation as e.g., why, why-not, what-if, how-to explanations

clarify the process of reasoning (i.e. increasing transparency and understandability). As even humans are sometimes insecure about judging the dialog partner and to decide whether and which type of reaction would be appropriate, it seems valid that a technical system will not overcome this issue of uncertainty. Therefore, we assume that the transfer of this problem to a technical system can only be handled effectively by incorporating uncertainty and thus using a probabilistic model. In the remainder of this paper, we will first elaborate how to react to not understandable situations and secondly present how to incorporate these findings into a multimodal dialogue system using a probabilistic model.

2 Coping with Incomprehensible Situations

Analogous to human-human interaction providing explanations in not understandable situations in HCI can reduce the loss of trust (Glass et al., 2008). However, HCT is not a one-dimensional simple concept. It may be devided into several components, which all have to be well-functioning to have the user trust a technical system. Existent studies concentrated on showing that explanations or different kinds of explanations can influence HCT in general (Lim et al., 2009). So, what is lacking currently is which explanations do influence which bases of human-computer trust.

2.1 Explanations

In general, explanations are given to clarify, change or impart knowledge. Usually the implicit idea consists of aligning the mental models of the participating parties. The mental model is the perceived representation of the real world, or in our case of the technical system and its underlying processes. In this context explanations try to establish a common ground between the parties in the sense that the technical system tries to clarify its actual model to the user. This is the attempt of aligning the user's mental model to the actual system. However, explanations do not always have the goal of aligning mental models, but can be used for other purposes as well. Analogous to human-human interaction, in human-computer interaction the sender of the explanation pursues a certain goal, with respect to the addressee, which should be achieved. The question remains, how these different goals of explanation (see table 1) map to HCT, meaning, how they influence HCT or components of it.

2.2 Human-Computer Trust

Mayer et al. (1995) define trust in human-human interaction to be ”the extent to which one party is willing to depend on somebody or something, in a given situation with a feeling of relative security, even though negative consequences are possible”. For HCI trust can be defined as ”the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability” (Lee and See, 2004). Technical Systems which serve as intelligent assistants with the purpose of helping the user in complex as well as in critical situations seem to be very dependent on an intact HCT relationship. However, trust is multi-dimensional and consists of several bases. For human relationships, Mayer et al. defined three levels that build the bases of trust: ability, integrity and benevolence. The same holds for HCI, where HCT is a composite of several bases. For human-computer trust Madsen and Gregor (2000) constructed a hierarchical model (see figure 1) resulting in five basic constructs or so-called bases of trust, which can be divided in two general components, namely cognitive-based and affect-based bases. In short-term human-computer interaction, cognitive-based HCT components seem to be more important, because it will be easier to influence those. Perceived understandability can be seen in the sense that the human supervisor or observer can form a mental model and predict future system behavior. The perceived reliability of the system, in the usual sense of repeated, consistent functioning. And technical competence means that the system is perceived to perform the tasks accurately and correctly based on the input information. In this context it is important to mention, that as Mayer already stated, the bases of trust are separable, yet related to one another. All bases must be perceived highly for the trustee to be

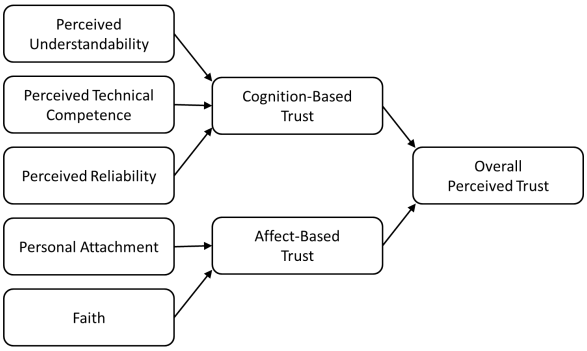


Figure 1: Human-computer trust model: Personal attachment and faith build the bases for affect-based trust. Rerceived understandability, technical competence and reliability for cognition-based trust.

deemed trustworthy. If any of the bases does not fulfill this requirement, the overall trustworthiness can suffer (Madsen and Gregor, 2000).

3 Related Work

Previous work on handling trust in technical systems was done for example by Glass et al. (2008). They investigated factors that may change the level of trust users are willing to place in adaptive agents. Among these verified findings were statements like "provide the user with the information provenance for sources used by the system", "intelligently modulating the granularity of feedback based on context- and user-modeling" or "supply the user with access to information about the internal workings of the system". However, what is missing in Glass et al.'s work is the idea of rating the different methods to uphold HCT in general and the use of a complex HCT model. Other related work was for example done by Lim et al. (2009) on how different kinds of explanations can improve the intelligibility of context-aware intelligent systems. They concentrate on the effect of Why, Why-not, How-to and What-if explanations on trust and understanding system's actions or reactions. The results showed that Why and Why-not explanations were the best kind of explanation to increase the user's understanding of the system, though trust was only increase by providing Why explanations. Drawbacks of this study were that they did only concentrate on understanding the system and trusting the system in general and did not consider that HCT is on the one hand not only influenced by the user's understanding of the system and on the other hand that if one base of trust is flawed, the HCT in general will be damaged (Mayer et al., 1995).

Regarding the issue of trusting a technical system or its actions and reactions related work exists for example on "credibility" (Fogg and Tseng, 1999). However, this term developed in the web community focusing on the believability of external sources. The term trust is used in the web research community as well as in work on "trust in automation". However, as Fogg stated himself later (Tseng and Fogg, 1999) credibility should be called believability and trust-in-automation should be called dependability to reduce the missunderstandings. In this work we use the term human-computer trust and its model by Madsen and Gregor (2000) subsuming both terms.

4 Experiment on Explanation Effectiveness

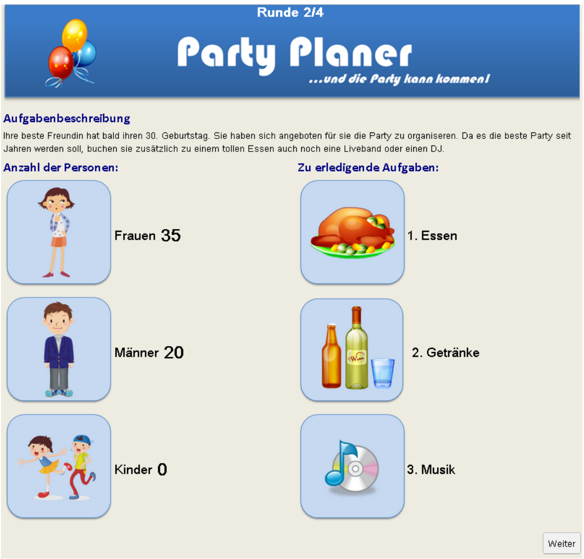
The insight that human-computer trust is not a simple but complex construct and the lack of directed methods to influence components of HCT motivated us to conduct an experiment which tried to overcome some of these issues. The use of explanations to influence HCT bases in a directed and not arbitrary way, depends on whether an effective mapping of explanation goals to HCT bases can be found. This means, that we have to identify which goal of explanation influences which base of trust in the most effective way. Therefore, the goal was to change undirected strategies to handle HCT issues into directed and well-founded ones, substantiating the choice and goal of explanation.

For that we conducted a web-based study inducing events to create not understandable or not expected situations and then compared the effects of the different goals of explanations on the HCT-bases. For our experiment we concentrated on justification and transparency explanations. Justifications are the most obvious goal an explanation can pursue. The main idea of this goal is to provide support for and increase confidence in given system advices or actions. The goal of transparency is to increase the users understanding in how the system works and reasons. This can help the user to change his perception of the system from a black-box to a system the user can comprehend. Thereby, the user can build a mental model of the system and its underlying reasoning processes.

The participants in the experiment where acquired by using flyers in the university as well as through facebook. The age of the participants was in a range from 14 to 61, with the mean being 24,1. Gender wise, the distribution was 59% (male) to 41% (female), with most of the participants being students. For the participation the students did receive a five euro voucher for a famous online store. However, this was only granted when finishing the complete experiment. Therefore, participants dropping out of the experiment would waive the right on the voucher.

4.1 Set-Up

The main objective of the participants to organize four parties for friends or relatives in a web-based environment. This means that they had to use the browser at home or the university to organize for example, the music, select the type and amount of food or order drinks. Each party was described by an initial screen depicting the key data for the party. This included which tasks had to be accomplished and how many people were expected to join (see figure 2). Each task was implemented as a single web-page, with the goal to organize one part of the party (i.e., dinner, drinks, or champagne reception). The user had to choose from several drop-down menus which item should be ordered for the party and in what number. For example, the user had to order the components of the dinner (see figure 3). When an entry inside a drop-down menu was chosen, the system gave an advice on how much of this would be needed to satisfy the needs of one guest. Additionally, before the participant could move on to the next task, the orders were checked by the system. The system would output whether the user had selected too much, too little or the right amount and only if everything was alright could proceed to the next task. The experiment consisted in total of four rounds. The first two rounds were meant to go smoothly and were supposed to get the subject used to the system and by that building a mental model of it. After the first two rounds a HCT questionnaire was presented to the user. As expected the user has built a relationship with the system by gaining an understanding of the systems processes. The next two rounds were meant to influence the HCT-relationship negative with unexpected external events. These unexpected, and incongruent to the user's mental model, system events were influencing pro-actively the decisions



Runde 2/4

Party Planer

...und die Party kann kommen!

Aufgabenbeschreibung

Ihre beste Freundin hat bald ihren 30. Geburtstag. Sie haben sich angeboten für sie die Party zu organiseren. Da es die beste Party seit Jahren werden soll, buchen sie zusätzlich zu einem tollen Essen auch noch eine Liveband oder einen DJ.

Anzahl der Personen:

Frauen 35

Männer 20

Kinder 0

Zu erledigende Aufgaben:

1. Essen

2. Getränke

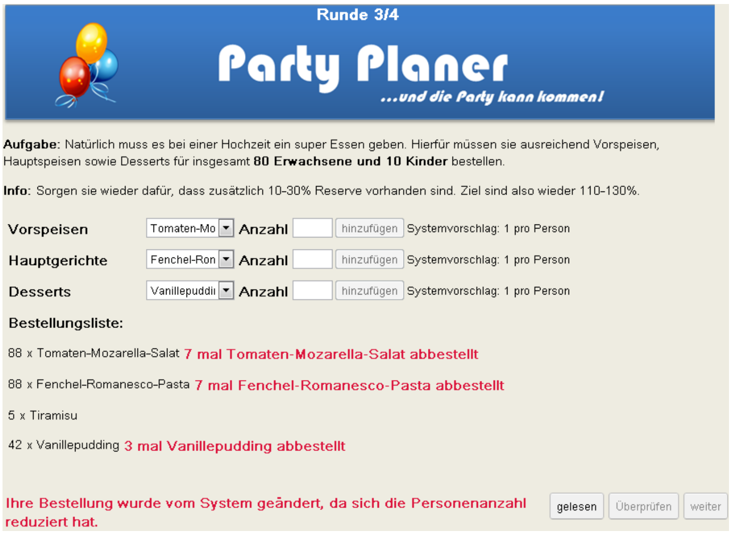
3. Musik

Figure 2: General information on the party. How many people plan to attend the event and what type of tasks have to be accomplished.

and solutions the user made to solve the task. This means, without warning, the user was overruled by the system and either simply informed by this change, or was presented an additional justification or transparency explanation as seen in figure 3. In this figure we can see that the user's order ('Bestellungsliste') was changed pro-actively because of an external event. Here the attendance of some participants was cancelled in the reservation system, thus the system did intervene. This pro-active change was explained at the bottom of the web-page by, in this case, providing a justification ('The order was changed by the system, because the number of attending persons decreased'). The matching transparency explanation would not only provide a reason, but explain how the system answer was reached ('Due to recent events the order was changed by the system. The order volume has been reduced, because several persons canceled their attendance in the registration system.'). Events like this occurred several times in the rounds 3 and 4 of the party planning.

4.2 Results

139 starting participants were distributed among the three test groups (no explanation, transparency, justifications). 98 accomplished round 2, reaching the point until the external events were induced and 59 participants completed the experiment. The first main result was that 47% from the group receiving no explanations quit during



Aufgabe: Natürlich muss es bei einer Hochzeit ein super Essen geben. Hierfür müssen sie ausreichend Vorspeisen, Hauptspeisen sowie Desserts für insgesamt **80 Erwachsene und 10 Kinder** bestellen.

Info: Sorgen sie wieder dafür, dass zusätzlich 10-30% Reserve vorhanden sind. Ziel sind also wieder 110-130%.

Vorspeisen	Tomaten-Mo	Anzahl	Systemvorschlag: 1 pro Person
Hauptgerichte	Fenchel-Ron	Anzahl	Systemvorschlag: 1 pro Person
Desserts	Vanillepuddi	Anzahl	Systemvorschlag: 1 pro Person

Bestellungsliste:

88 x Tomaten-Mozarella-Salat 7 mal Tomaten-Mozarella-Salat abbestellt

88 x Fenchel-Romanesco-Pasta 7 mal Fenchel-Romanesco-Pasta abbestellt

5 x Tiramisu

42 x Vanillepudding 3 mal Vanillepudding abbestellt

Ihre Bestellung wurde vom System geändert, da sich die Personenanzahl reduziert hat.

Figure 3: This screenshot shows one of the tasks the user has to accomplish. In this case dinner ('Hauptgerichte') including entree ('Vorspeisen') and desserts has to be ordered.

the critical rounds 3 and 4. However, if explanations were presented only 33% (justifications) and 35% (transparency) did quit. This means that eventhough the participants would encounter negative consequences of losing the reward money, they did drop out of the experiment. Therefore, we can state that the use of explanations in incomprehensible and not expected situations can help to keep the human-computer interaction running. The main results from the HCT-questionnaires can be seen in figure 4. The data states that providing no explanations in rounds three and four resulted in a decrease in several bases of trust. Therefore, we can conclude that the external events did indeed result in our planned negative change in trust. *Perceived understandability* diminished on average over the people questioned by 1.2 on a Likert scale with a range from 1 to 5 when providing no explanation at all compared to only 0.4 when providing *transparency* explanations (no explanation vs. transparency t(34)=-3.557 $p<0.001$), and on average by 0.5 with *justifications* (no explanation vs. justifications t(36)=-2.023 $p<0.045$). Omitting explanations resulted in an average decrease of 0.9 for the *perceived reliability*, with transparency explanations in a decrease of 0.4 and for *justifications* in a decrease of 0.6 (no explanation vs. transparency t(34)=-2.55 $p<0.015$).

These results support our hypotheses that transparency explanations can help to reduce the negative effects of trust loss regarding the user's perceived understandability and reliability of the system in incomprehensible and unexpected situations. Especially for the base of understandability, meaning the prediction of future outcomes, transparency explanations fulfill their purpose in a good way. Additionally, they seem to help with the perception of a reliable, consistent system. The results show that it is worthwhile to augment ongoing dialogs with explanations to maintain HCT.

While analyzing the data we did not find any statistically significant differences between providing transparency and justification explanations. However, this could be due to limited differences in the goals of explanation. Usually, the transparency explanations in the experiment were including more information on what happened inside the system, and how the system did recognize the external event (e.g., the reduction of attending persons). In future experiments we will try to distinguish those two goals of explanations more from each other. For example, the justification for reduce attendance to an event can be changed to something like 'The order was changed by the system, because otherwise you would have too much food' instead of 'The order was changed by the system, because the number of attending persons decreased' and by that making it more different from the transparency explanation ('Due to recent events the order was changed by the system. The order volume has been reduced, because several persons canceled their attendance in the registration system.'). In the following, we will describe how this is used in our developed explanation aug-

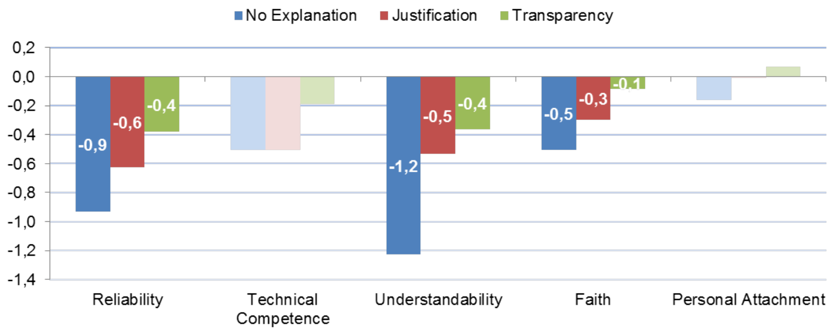


Figure 4: This figure shows the changes of HCT bases from round 2 to round 4. The scale was a 5 point likert scale with e. g., 1 the system being not understandable at all and 5 the opposite.

mentation architecture (see figure 5).

5 Implementation

The augmentation of the dialog is done using two different kinds of dialog models. On the one hand we are using a classic dialog model based on a finite-state machine approach for the task-oriented part of the dialog. On the other hand a planner (Müller et al., 2012) is used to generate from a POMDP a decision tree. This POMDP is used only for the augmentation of the task-oriented part of the dialog with explanations. The communication between each module of the architecture is controlled by a XML-based message-oriented middleware (Schröder, 2010), using a publish-subscribe system to distribute the XML-messages. In order to decide when to induce additional explanations, on one hand critical situations in HCI have to be recognized and on the other hand, if necessary the appropriate type of explanation has to be given. Obviously, recognizing those situations cannot be done solely by using information coming from interaction and its history. Multimodal input as speech recognition accuracy, facial expressions or any other sensor information can help to improve the accuracy of recognizing critical moments in HCI. However, mapping sensor input to semantic information is usually done by classifiers and those classifiers convey a certain amount of probabilistic inaccuracy which has to be handled. Therefore, a decision model has to be able to handle probabilistic information in a suitable manner.

5.1 Probabilistic Decision Model

For the problem representation when and how to react, a so-called partially observable Markov decision process (POMDP) was chosen and formalized in the Relational Dynamic Influence Diagram Language (RDDL) (Sanner, 2010). RDDL is a uniform language which allows an efficient description of POMDPs by representing its constituents (actions, observations, belief state) with variables. Formally, a POMDP consists of a set S of world states, a set A of system actions, and a set O of possible observations the system can make. Further, transition probabilities $P(s'|s,a)$ describe the dynamics of the environment, i.e., the probability of the successor world state being s' when action a is executed in state s. The observation probabilities $P(o|s',a)$ represent the sensors of the system in terms of the probability of making observation o when executing a resulted in successor world state s'. Each time the system executes an action a, it receives a reward $R(s,a)$ which depends on the world state s the action was executed in. The overall goal of the system is to maximize the accumulated reward it receives over a fixed number of time steps. (For more information on POMDPs, see Kaelbling et al. (1998).)

A POMDP is then used by a planner (Silver and Veness, 2010; Müller et al., 2012) to search for a policy that determines the system's behavior. This policy is, e.g., represented as a decision tree that recommends the most suitable action based on the system's previous actions and observations. For example, a policy for a POMDP that models HCI with respect to HCT, can thus represent a decision tree which represents a guideline for a dialog flow which ensures an intact HCT-relationship.

The RDDL model is a probabilistic representation of the domain, which determines when and how to augment the dialog with explanations at run-time. Each observation o consists of the du-

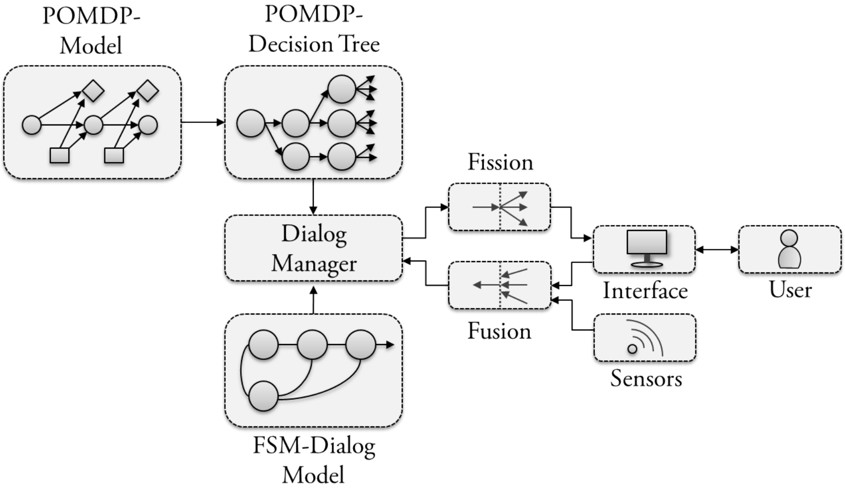


Figure 5: The architecture consists of two dialog models, a fission and fusion engine, sensors as well as the multimodal interface representation to interact with the user. The dialog models can be seperated in a task-oriented FSM-dialog model and into a POMDP-based decision tree for explanation augmentation. This decision tree is generated from a POMDP-model by a planner.

ration of interaction for each dialog step as well as the semantic information of the input (i.e., which action in the interface was triggered by speech, touch or point-and-click interaction). Those types of interaction can bring along uncertainty (e.g., speech recognition rate). The state s in terms of HCT is modeled by its respective bases, namely understandability, technical-competence, reliability, faith and personal attachment. The system actions A are the dialogs presented to the user. These are the different goals of explanations (justification, transparency, conceptualization, relevance and learning) as well as the task-oriented part of the dialog represented by a so-called *communicative function(c)* with c from set C (e.g., question, inform, answer, offer, request, instruct). This means, that in the POMDP only the communicative function of the task-oriented dialogs is represented without the specific content.

The transition probabilities are defined as *conditional probability functions* (CPFs) and model user behavior dependent on the system's actions and the user's current HCT values. Basically, conditional functions are defined using *if else* for all wanted cases. For example, we defined that the user's understanding in s' will probably be high if a transparency explanation was the last system action. When the user's understanding is indeed high in s', the observation will probably be that the user clicked *okay*, and the time he took for the interaction was around his usual amount taken for explanations. From this observation, a planner can infer that the transparency explanation indeed increased the user's understanding.

Now, the quest is to define the reward function $R(s, a)$ in a way that it leads to an optimal flow of actions. I.e., the system should receive a penalty when the bases of trust do not remain intact, and actions should incur a cost so that the system only executes them when trust is endangered. However, because POMDPs tend to be become very quick very complex, we chose to seperate the task-oriented dialog from the additional dialog augmentation with explanations when needed.

5.2 Dialog Augmentation Process

The task-oriented dialog is modeled as a classic finite-state machine (FSM). Each dialog action has several interaction possibilities, each leading to another specified dialog action. Each of those dialog action is represented as POMDP action a as part of C (*communicative function(c)*). As already mentioned, only the communicative function is modeled to reduce the complexity in the POMDP.

The HCI is started using the FSM-based dialog model approach and uses the POMDP to check whether the user's trust or components of the user's trust are endangered. At run-time the next action in the FSM is compared to the one determined by the POMDP (see figure 6). This means, that if the next action in the FSM is not the same as the one planned by the POMDP, the dia-

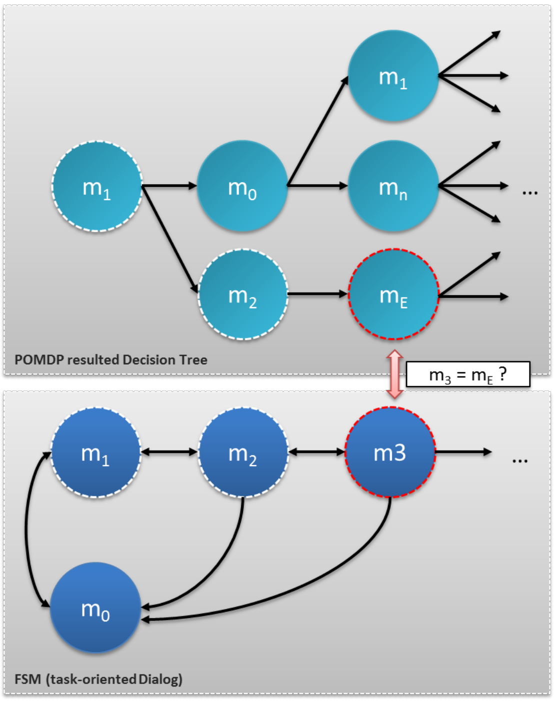


Figure 6: This figure shows the comparison of FSM to Decision Tree. The next action m_3 in the FSM does not correspond to the one endorsed by the POMDP Decision Tree. Therefore, the dialog will be augmented by explanation action m_E.

log flow is interrupted, and the ongoing dialog is augmented by the proposed explanation. For example, if the user is presented currently a communicative function of type *inform* and the decision tree recommends to provide a transparency explanation, because the understanding and reliability are probably false, the originally next step in the FSM is postponed and first the explanation is presented. The other way around, if the next action in the FSM is subsumed by the one scheduled by the POMDP, the system does not need to intervene. For example, if the next FSM-action is to instruct the user about how to connect *amplifier* and *receiver* and the POMDP would recommend an action of type communicative function *instruct*, no dialog augmentation is needed.

6 Dialog Interface

Each dialog action in the FSM as well as the explanation dialogs are represented by a so-called dialog goal, which is allocated on the one hand a type of communicative function c. On the other hand the dialog content is composed of multiple information objects referencing so-called *information IDs* in the information model. Each information object can consist of different types (e.g., text, audio, and pictures). For interface presentation the dialog goal is passed to the fission which selects and combines the information objects at runtime by a fission sub-component to compose the user interface in a user- and situation-adaptive way (Honold et al., 2012). In figure 7 we can see a typical interface for a transmitted dialog goal in which the user can interact via speech, touch or GUI.

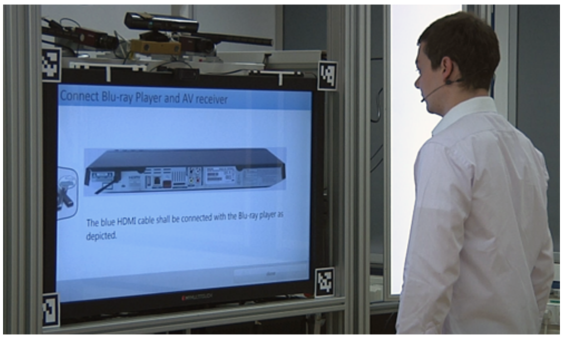

Connect Blu-ray Player and AV receiver
The blue HDMI cable shall be connected with the Blu-ray player as depicted.

Figure 7: A typical output presentation of the fission component of a dialog goal. Here the user gets instruction on how to connect the *BluRay-Player* with an *HDMI* cable.

7 Conclusion and Future Work

In this paper we showed the necessity to deal with critical situations in HCI in a probabilistic approach. The advantage of our approach is that the designer still can define a FSM-based task-oriented dialog. Usually most commercial systems are still based on such approaches. However, expanding the dialog by a probabilistic decision model seems to be a valuable choice. Our experiment on the influence of explanations on HCT has clearly shown, that it is worthwhile to augment the ongoing dialog by transparency or justification explanations for an intact HCT relationship. In the future we will run experiments on how effective the hybrid FSM-POMDP approach is compared to classic as well as POMDP dialog systems.

Acknowledgment

This work was supported by the Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” which is funded by the German Research Foundation (DFG).

References

B. J. Fogg and Hsiang Tseng. 1999. The elements of computer credibility. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '99, pages 80–87, New York, NY, USA. ACM.

Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. 2008. Toward establishing trust in adaptive agents. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 227–236, NY, USA. ACM.

Frank Honold, Felix Schüssel, and Michael Weber. 2012. Adaptive probabilistic fission for multimodal systems. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 222–231.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, pages 99–134.

John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2119–2128, NY, USA. ACM.

Maria Madsen and Shirley Gregor. 2000. Measuring human-computer trust. In *Proceedings of the 11 th Australasian Conference on Information Systems*, pages 6–8.

Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An Integrative Model of Organizational Trust. *The Academy of Management Review*, 20(3):709–734.

B M Muir. 1992. Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. In *Ergonomics*, pages 1905–1922.

Felix Müller, Christian Späth, Thomas Geier, and Susanne Biundo. 2012. Exploiting expert knowledge in factored POMDPs. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, pages 606–611.

Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2):230–253, June.

Scott Sanner. 2010. Relational dynamic influence diagram language (rddl): Language description. http://users.cecs.anu.edu.au/ ssanner/IPPC2011/RDDL.pdf.

Marc Schröder. 2010. The semaine api: Towards a standards-based framework for building emotion-oriented systems. *Advances in Human-Machine Interaction*, (319406):21.

D. Silver and J. Veness. 2010. Monte-carlo planning in large POMDPs. In *NIPS*, pages 2164–2172.

F. Sørmo and J. Cassens. 2004. Explanation goals in case-based reasoning. In *Proceedings of the 7th European Conference on Case-Based Reasoning*, pages 165–174.

Shawn Tseng and B. J. Fogg. 1999. Credibility and computing technology. *Commun. ACM*, 42(5):39–44, May.